Package: APackOfTheClones (via r-universe)

October 15, 2026

Type Package

Title Visualization of Clonal Expansion for Single Cell Immune Profiles

Version 1.3.0

Maintainer Qile Yang <qile.yang@berkeley.edu>

Description Visualize clonal expansion via circle-packing. 'APackOfTheClones' extends 'scRepertoire' to produce a publication-ready visualization of clonal expansion at a single cell resolution, by representing expanded clones as differently sized circles. The method was originally implemented by Murray Christian and Ben Murrell in the following immunology study: Ma et al. (2021) <doi:10.1126/sciimmunol.abg6356>.

BugReports https://github.com/Qile0317/APackOfTheClones/issues

Depends R (>= 4.1.0)

License MIT + file LICENSE

Encoding UTF-8

Language en-US

LazyData false

Suggests cowplot, covr, knitr, rmarkdown, spelling, testthat (>= 3.0.0), vdiffr, colorspace

Config/testthat/edition 3

Imports assertthat, dplyr, ggforce, ggplot2, hash, lifecycle, magrittr, methods, Rcpp, rlang, Seurat, SeuratObject

RoxygenNote 7.3.2

URL https://qile0317.github.io/APackOfTheClones/, https://github.com/Qile0317/APackOfTheClones, https://joss.theoj.org/papers/10.21105/joss.06868

LinkingTo Rcpp

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev python3 zlib1g-dev

Repository https://qile0317.r-universe.dev

Date/Publication 2025-08-28 21:36:44 UTC

RemoteUrl https://github.com/qile0317/apackoftheclones

RemoteRef HEAD

RemoteSha 19db788d9fe4c2392b5d74651f2a74b531d82b92

Contents

AdjustAPOTC . . . 2
APOTCPlot . . . 6
combined_pbmc . . . 10
containsApotcRun . . . 11
countCloneSizes . . . 12
deleteApotcData . . . 13
getApotcDataIds . . . 14
getLastApotcDataId . . . 15
getReductionCentroids . . . 16
getSharedClones . . . 17
mergeCloneSizes . . . 19
overlayLegend . . . 20
removeLegend . . . 22
renameApotcRun . . . 23
RunAPOTC . . . 24
showCloneHighlight . . . 28
updateApotc . . . 29
vizAPOTC . . . 30

Index **36**

`AdjustAPOTC` *Adjust the parameters of the APackOfTheClones reduction in a seurat object*

Description

[Stable]

If the user is unsatisfied with the clonal expansion plot that was generated from `RunAPOTC()` and `APOTCPlot()`, this function has a range of arguments to modify the data and/or parameters of the visualization. Note that some of the arguments may conflict with each other.

Usage

```
AdjustAPOTC(
  seurat_obj,
  reduction_base = NULL,
  clonecall = NULL,
  ...,
  extra_filter = NULL,
  run_id = NULL,
  new_rad_scale_factor = NULL,
  new_clone_scale_factor = NULL,
  repulse = FALSE,
  repulsion_threshold = 1,
  repulsion_strength = 1,
  max_repulsion_iter = 10L,
  relocate_cluster = NULL,
  relocation_coord = NULL,
  nudge_cluster = NULL,
  nudge_vector = NULL,
  recolor_cluster = NULL,
  new_color = NULL,
  rename_label = NULL,
  new_label = NULL,
  relocate_label = NULL,
  label_relocation_coord = NULL,
  nudge_label = NULL,
  label_nudge_vector = NULL,
  verbose = TRUE
)
```

Arguments

`seurat_obj`	The seurat object to be adjusted.
`reduction_base`	character. The seurat reduction to base the clonal expansion plotting on. Defaults to `'umap'` but can be any reduction present within the reductions slot of the input seurat object, including custom ones. If ''pca'', the cluster coordinates will be based on PC1 and PC2. However, generally APackOfTheClones is used for displaying UMAP and occasionally t-SNE versions to intuitively highlight clonal expansion.
`clonecall`	character. The column name in the seurat object metadata to use. See `scRepertoire` documentation for more information about this parameter that is central to both packages.
`...`	additional "subsetting" keyword arguments indicating the rows corresponding to elements in the seurat object metadata that should be filtered by. E.g., `seurat_clusters = c(1, 9, 10)` will filter the cells to those in the `seurat_clusters` column with any of the values 1, 9, and 10. Unfortunately, column names in the seurat object metadata cannot conflict with the keyword arguments. **MAJOR NOTE** if any subsetting keyword arguments are a *prefix* of any preceding argument names

(e.g. a column named `reduction` is a prefix of the `reduction_base` argument) R will interpret it as the same argument unless *both* arguments are named. Additionally, this means any subsequent arguments *must* be named.

`extra_filter` character. An additional string that should be formatted *exactly* like a statement one would pass into dplyr::filter that does *additional* filtering to cells in the seurat object - on top of the other keyword arguments - based on the metadata. This means that it will be logically AND'ed with any keyword argument filters. This is a more flexible alternative / addition to the filtering keyword arguments. For example, if one wanted to filter by the length of the amino acid sequence of TCRs, one could pass in something like `extra_filter = "nchar(CTaa) - 1 > 10"`. When involving characters, ensure to enclose with single quotes.

`run_id` character. This will be the ID associated with the data of a run, and will be used by other important functions like `APOTCPlot()` and AdjustAPOTC. Defaults to `NULL`, in which case the ID will be generated in the following format:
`reduction_base;clonecall;keyword_arguments;extra_filter`
where if keyword arguments and extra_filter are underscore characters if there was no input for the `...` and `extra_filter` parameters.

`new_rad_scale_factor`
a single numeric in (0, 1]. changes the radius scaling factor of all circles.

`new_clone_scale_factor`
a single numeric in (0, 1]. changes the clone_scale_factor

`repulse` If `TRUE`, will attempt to push overlapping clusters away from each other.

`repulsion_threshold`
numeric. The radius that clonal circle clusters overlap is acceptable when repulsing.

`repulsion_strength`
numeric. The smaller the value the less the clusters repulse each other per iteration, and vice versa.

`max_repulsion_iter`
integer. The number of repulsion iterations.

`relocate_cluster`
Numeric or Character. Indicates which cluster(s) based on the index or label to relocate to new coordinates.

`relocation_coord`
numeric of length two or a list of numerics of length two of length of `relocate_cluster`. If its a list, indicates each coordinate that the clusters in `relocate_cluster` should move to. If its just a numeric, then will relocate all clusters in `relocate_cluster` to the input, which is likely not desired behavior, so this should only be convenience syntax if `relocate_cluster` has length 1.

`nudge_cluster` Numeric or Character. Indicates which cluster(s) based on the index or label to "nudge"/translate their coordinate(s) by.

`nudge_vector` numeric of length two or a list of numerics of length two of length of `nudge_cluster`. If its a list, indicates each translation vector (in other words, x-y coordinates) that the clusters in `nudge_cluster` should be translate by. If its just a numeric, then will translate all clusters in `nudge_cluster` by the input - which mostly is syntactic sugar for translating a single cluster if the input of `nudge_cluster` is of length 1.

recolor_cluster
: Numeric or Character. Indicates which cluster(s) based on the index or label to change their color by.

new_color
: character of arbitrary length. Indicates the corresponding new colors that selected clusters in recolor_cluster should be changed to.

rename_label
: Numeric or character. Indicates the index or name of label(s) to be renamed.

new_label
: Character. Indicates the corresponding new label(s) that selected label(s) in rename_label should be changed to.

relocate_label
: Numeric or character. Indicates the index or name of label(s) to be relocated.

label_relocation_coord
: Numeric of length two or a list of numerics of length two of length of relocate_label. If it's a list, indicates each coordinate that the labels in relocate_label should move to. If it's just a numeric, then will relocate all labels in relocate_label to the input, which is likely not desired behavior, so this should only be convenience syntax if relocate_label has length 1.

nudge_label
: Numeric or character. Indicates the index or name of label(s) to be "nudged"/translated.

label_nudge_vector
: Numeric of length two or a list of numerics of length two of length of nudge_label. If it's a list, indicates each translation vector (in other words, x-y coordinates) that the labels in nudge_label should be translated by. If it's just a numeric, then will translate all labels in nudge_label by the input - which mostly is syntactic sugar for translating a single label if the input of nudge_label is of length 1.

verbose
: logical. Decides if visual cues are displayed to the R console of the progress.

Value

The adjusted seurat_obj

Examples

```
# do an APackOfTheClones run
pbmc <- RunAPOTC(get(data("combined_pbmc")), verbose = FALSE)

# adjust the rad_scale_factor, and nudge cluster 1 by x = 1, y = 1
pbmc <- AdjustAPOTC(
    pbmc,
    new_rad_scale_factor = 0.9,
    nudge_cluster = 1,
    nudge_vector = c(1, 1),
    verbose = FALSE
)

# plot the result
APOTCPlot(pbmc)

# rename some labels
pbmc <- AdjustAPOTC(
```

```
    pbmc, rename_label = c(2, 5), new_label = c("Foo", "Bar")
)

# perhaps multiple clusters need to be relocated and repulsed
pbmc <- AdjustAPOTC(
    pbmc,
    relocate_cluster = c("Foo", "C10"), # using labels works too
    relocation_coord = list(c(2, 3.5), c(0, 5)),
    repulse = TRUE,
    verbose = FALSE
)

# plot again to check results
APOTCPlot(pbmc, show_labels = TRUE, verbose = FALSE)
```

`APOTCPlot` *Various variations of visualizations of clonal expansion post-RunAPOTC*

Description

[Stable]

Given a seurat object with an 'apotc' (APackOfTheClones) object from running `RunAPOTC()`, this function will read the information and return a customizable ggplot2 object of the clonal expansion with a circle size legend. If the user is unhappy about certain aspects of the plot, many parameters can be adjusted with the AdjustAPOTC function.

The specific APackOfTheClones run to be plotted can be identified in two ways: either by inputting the `run_id` associated with the run that was either defined / auto-generated during `RunAPOTC()`, or by inputting the `reduction_base`, `clonecall`, `extra_filter` and any other keyword arguments that corresponded to the run. Its heavily recommended to use the `run_id`. If none of these parameters are inputted, the function defaults to returning the plot of the latest run.

Usage

```
APOTCPlot(
  seurat_obj,
  reduction_base = NULL,
  clonecall = NULL,
  ...,
  extra_filter = NULL,
  run_id = NULL,
  show_shared = NULL,
  only_link = NULL,
  show_all_links = FALSE,
  clone_link_width = "auto",
  clone_link_color = "black",
  clone_link_alpha = 0.5,
```

```
  res = 360L,
  linetype = "blank",
  use_default_theme = TRUE,
  retain_axis_scales = FALSE,
  alpha = 1,
  show_labels = FALSE,
  label_size = 5,
  add_size_legend = TRUE,
  legend_sizes = "auto",
  legend_position = "auto",
  legend_buffer = 0.2,
  legend_color = "#808080",
  legend_spacing = "auto",
  legend_label = "Clone sizes",
  legend_text_size = 5,
  add_legend_background = TRUE,
  add_legend_centerspace = 0,
  detail = TRUE,
  verbose = TRUE
)
```

Arguments

`seurat_obj`	A seurat object that has been integrated with clonotype data and has had a valid run of `RunAPOTC()`.
`reduction_base`	character. The seurat reduction to base the clonal expansion plotting on. Defaults to `'umap'` but can be any reduction present within the reductions slot of the input seurat object, including custom ones. If ''pca'', the cluster coordinates will be based on PC1 and PC2. However, generally APackOfTheClones is used for displaying UMAP and occasionally t-SNE versions to intuitively highlight clonal expansion.
`clonecall`	character. The column name in the seurat object metadata to use. See `scRepertoire` documentation for more information about this parameter that is central to both packages.
`...`	additional "subsetting" keyword arguments indicating the rows corresponding to elements in the seurat object metadata that should be filtered by. E.g., `seurat_clusters = c(1, 9, 10)` will filter the cells to those in the `seurat_clusters` column with any of the values 1, 9, and 10. Unfortunately, column names in the seurat object metadata cannot conflict with the keyword arguments. **MAJOR NOTE** if any subsetting keyword arguments are a *prefix* of any preceding argument names (e.g. a column named `reduction` is a prefix of the `reduction_base` argument) R will interpret it as the same argument unless *both* arguments are named. Additionally, this means any subsequent arguments *must* be named.
`extra_filter`	character. An additional string that should be formatted *exactly* like a statement one would pass into dplyr::filter that does *additional* filtering to cells in the seurat object - on top of the other keyword arguments - based on the metadata. This means that it will be logically AND'ed with any keyword argument filters. This is a more flexible alternative / addition to the filtering keyword arguments.

For example, if one wanted to filter by the length of the amino acid sequence of TCRs, one could pass in something like `extra_filter = "nchar(CTaa) - 1 > 10"`. When involving characters, ensure to enclose with single quotes.

`run_id` character. This will be the ID associated with the data of a run, and will be used by other important functions like `APOTCPlot()` and AdjustAPOTC. Defaults to `NULL`, in which case the ID will be generated in the following format: `reduction_base;clonecall;keyword_arguments;extra_filter`

where if keyword arguments and extra_filter are underscore characters if there was no input for the `...` and `extra_filter` parameters.

`show_shared` The output of getSharedClones can be inputted here, and the resulting plot will overlay lines between clone circles if that clonotype is common between clusters. Note that the input **must** be generated from data in the correct `APackOfTheClones` run, and the behavior is undefined otherwise and will likely error. The next 4 arguments allow for aesthetic customization of these line links.

`only_link` Optional integer indicating to only display clone links originating from this cluster if showing shared clones.

`show_all_links` **[Experimental]** logical. If `TRUE`, will show all links between clusters. Defaults to `FALSE` which only shows links between some clusters without overlapping, although there is currently no way to control which.

`clone_link_width`

numeric. The width of the lines that connect shared clones. Defaults to `"auto"` which will estimate a reasonable value depending on circle sizes.

`clone_link_color`

character. The color of the lines that connect shared clones. Defaults to `"blend"` which will use the average colors of the two connected clones. Else, any hex color or valid color string input will work, and the corresponding color will be applied on all links.

`clone_link_alpha`

numeric. The alpha of the lines that connect shared clones.

`res` The number of points on the generated path per full circle. From plot viewers, if circles seem slightly too pixelated, it is recommended to first try to export the plot as an `.svg` before increasing `res` due to increased plotting times from ggforce::geom_circle.

`linetype` The type of outline each circle should have. defaults to `"blank` meaning no outline. More information is in the function documentation of `ggforce::geom_circle`.

`use_default_theme`

logical that defaults to `TRUE`. If `TRUE`, the resulting plot will have the same theme as the seurat reference reduction plot. Else, the plot will simply have a blank background.

`retain_axis_scales`

If `TRUE`, approximately maintains the axis scales of the original reduction plot. However, it will only attempt to extend the axes and never shorten. Users are recommended to set this to `TRUE` especially if working with subsetted versions of the clonal data to better preserve the geometric relation to the original dimensional reduction.

`alpha` numeric. The alpha of the circles in (0, 1]. Defaults to 1.

`show_labels` If `TRUE`, will label each circle cluster at the centroid, defaulting to "C0, C1, ...".

`label_size` The text size of labels if shown. Defaults to 5.

`add_size_legend`
If `TRUE`, adds a legend to the plot visualizing the relative sizes of clones. Note that it is simply an overlay and not a real ggplot2 legend.

`legend_sizes` numeric vector. Indicates the circle sizes to be displayed on the legend, and will always be sorted from smallest to greatest. Defaults to `"auto"` which estimate a reasonable range of sizes to display.

`legend_position`
character or numeric. Can be set to either `"top_left"`, `"top_right"`, `"bottom_left"`, `"bottom_right"` and places the legend roughly in the corresponding position. Otherwise, can be a numeric vector of length 2 indicating the x and y position of the *topmost (smallest) circle* of the legend.

`legend_buffer` numeric. Indicates how much to "push" the legend towards the center of the plot from the selected corner. If negative, will push away

`legend_color` character. Indicates the hex color of the circles displayed on the legend. Defaults to the hex code for a gray tone

`legend_spacing` numeric. Indicates the horizontal distance between each stacked circle on the size legend. Defaults to `"auto"` which will use an estimated value depending on plot size

`legend_label` character. The title of the legend, which defaults to `"clone sizes`.

`legend_text_size`
numeric. The text size of the letters and numbers on the legend

`add_legend_background`
logical. If `TRUE`, will add a border around the legend and fill the background to be white, overlaying anything else.

`add_legend_centerspace`
numeric. An additional amount of distance changed between the circle sizes on the left side of the legend and the numbers on the right. Useful to set to around 0.5 (or more / less) when there are particularly large clone sizes that may cover the numbers.

`detail` logical. If `FALSE`, will only plot entire clusters as one large circle, which may be useful in cases where there are a high number of clones resulting in a large number of circles on the resulting ggplot, which has increased plotting times, and certain aspects of the plot needs to be finely adjusted with AdjustAPOTC or simply inspected. This should not be set to `FALSE` for the actual clonal expansion plot.

`verbose` logical. Decides if visual cues are displayed to the R console of the progress.

Value

A ggplot object of the APackOfTheClones clonal expansion plot of the seurat object. There is an additional 10th element in the object named `"APackOfTheClones"` used by other functions in this package and shouldn't interfere with any other ggplot functionality. (As far as currently known)

See Also

AdjustAPOTC

Examples

```
data("combined_pbmc")

combined_pbmc <- RunAPOTC(
    combined_pbmc, run_id = "run1", verbose = FALSE
)

# plotting with default arguments will plot the latest "run1"
clonal_packing_plot <- APOTCPlot(combined_pbmc)
```

---

combined_pbmc *Example Multi-sampled T-cell seurat object with integrated TCR library*

Description

[Stable]

Generated with `scRepertoire::combineExpression`. To construct this object from scratch, try:

scRepertoire::combineExpression(scRepertoire::combineTCR(get(data("contig_list", , package = "scRepertoire")), samples = c("P17B", "P17L", "P18B", , "P18L", "P19B", "P19L", "P20B", "P20L"), removeNA = FALSE, , removeMulti = FALSE, filterMulti = FALSE), get(data("scRep_example", , package = "scRepertoire")), cloneCall = "gene", proportion = TRUE)

Usage

```
data("combined_pbmc")
```

Format

A Seurat object with the following slots filled

assays Currently only contains one assay ("RNA" - scRNA-seq expression data)

counts - Raw expression data

- data - Normalized expression data
- scale.data - Scaled expression data
- var.features - names of the current features selected as variable
- meta.features - Assay level metadata such as mean and variance

meta.data Cell level metadata with a combined TCR contig list `from scRepertoire`

active.assay Current default assay

active.ident Current default idents

graphs Neighbor graphs computed, currently stores the SNN

reductions Dimensional reductions: UMAP

version Seurat version used to create the object

commands Command history, including the one used to create this object "combineExpression"

`containsApotcRun` *Check for the existence of an APackOfTheClones run with its run id*

Description

[Stable]

A convenience function to check for the existence of an APackOfTheClones run with its run id, regardless of if any run has been made

Usage

```
containsApotcRun(seurat_obj, run_id)
```

Arguments

`seurat_obj`	a seurat object
`run_id`	character. The id of the associated ApotcRun.

Value

A logical indicating whether the run exists.

Examples

```
pbmc <- RunAPOTC(
    seurat_obj = get(data("combined_pbmc")),
    reduction_base = "umap",
    clonecall = "strict",
    run_id = "run1",
    verbose = FALSE
)

containsApotcRun(pbmc, "run1")
#> [1] TRUE

containsApotcRun(pbmc, "run2")
#> [1] FALSE
```

`countCloneSizes` *count the number of clonotype sizes in a seurat object combined with a VDJ library overall or by cluster*

Description

[Stable]

Get clonotype frequencies from a seurat object's metadata, either as one whole table, or in a list of tables, based on the current / some custom ident of each cell. Note that depending on the ident (indicated by the `by_cluster` argument) there may be more or less clonotypes counted based on the number of rows containing NA for that column of that ident if it isn't the active ident.

Usage

```
countCloneSizes(
  seurat_obj,
  clonecall = "strict",
  extra_filter = NULL,
  ...,
  by_cluster = TRUE,
  sort_decreasing = NULL
)
```

Arguments

`seurat_obj`	a seurat object combined with a VDJ library with `scRepertoire`.
`clonecall`	character. The column name in the seurat object metadata to use. See `scRepertoire` documentation for more information about this parameter that is central to both packages.
`extra_filter`	character. An additional string that should be formatted *exactly* like a statement one would pass into dplyr::filter that does *additional* filtering to cells in the seurat object - on top of the other keyword arguments - based on the metadata. This means that it will be logically AND'ed with any keyword argument filters. This is a more flexible alternative / addition to the filtering keyword arguments. For example, if one wanted to filter by the length of the amino acid sequence of TCRs, one could pass in something like `extra_filter = "nchar(CTaa) - 1 > 10"`. When involving characters, ensure to enclose with single quotes.
`...`	additional "subsetting" keyword arguments indicating the rows corresponding to elements in the seurat object metadata that should be filtered by. E.g., `seurat_clusters = c(1, 9, 10)` will filter the cells to those in the `seurat_clusters` column with any of the values 1, 9, and 10. Unfortunately, column names in the seurat object metadata cannot conflict with the keyword arguments. **MAJOR NOTE** if any subsetting keyword arguments are a *prefix* of any preceding argument names (e.g. a column named `reduction` is a prefix of the `reduction_base` argument) R will interpret it as the same argument unless *both* arguments are named. Additionally, this means any subsequent arguments *must* be named.

by_cluster Logical or Character. If TRUE, will output a list of table objects, with the table at each index corresponding to level in Idents(). Each table's names are the clonotype name indicated by clonecall after filtering, while the values are the actual clone sizes. If FALSE, outputs just the aggregate clone sizes for all cells. Note that if FALSE, the output should be identical to that produced by mergeCloneSizes(countCloneSizes(..., by_cluster = TRUE)). Otherwise, this argument can also be a character indicating some column in the seurat object metadata to use a cell identity guiding (e.g. "seurat_clusters").

sort_decreasing a logical or NULL. If TRUE/FALSE, sorts each/the table by clonotype frequency with largest/smallest clones first with a stable sorting algorithm, and if NULL, no order is guaranteed but the output is deterministic.

Value

A list of tables or a single table depending on by_cluster

See Also

mergeCloneSizes

Examples

```
data("combined_pbmc")

countCloneSizes(combined_pbmc)
countCloneSizes(combined_pbmc, "aa")
countCloneSizes(combined_pbmc, "nt", orig.ident = c("P17B", "P17L"))
```

---

deleteApotcData *Delete the results of an APackOfTheClones run*

Description

[Stable]

A convenience function to erase all data associated with a particular run, including the ApotcData and command in seurat_obj@command. The run_id would be no longer accessible afterwards.

Usage

```
deleteApotcData(seurat_obj, run_id)
```

Arguments

seurat_obj a seurat object that has had RunAPOTC ran on it before in order of the functions being called.

run_id character. The id of the associated ApotcRun.

Value

The modified input seurat object

Examples

```
pbmc <- RunAPOTC(
    seurat_obj = get(data("combined_pbmc")),
    reduction_base = "umap",
    clonecall = "strict",
    run_id = "run1",
    verbose = FALSE
)

getApotcDataIds(pbmc)
#> [1] "run1"

# delete the data
pbmc <- deleteApotcData(pbmc, "run1")

getApotcDataIds(pbmc)
#> NULL
```

`getApotcDataIds` *Get all run ids of previous RunAPOTC runs on a seurat object*

Description

[Stable]

A convenience function to get all run ids of previous RunAPOTC run IDs

Usage

```
getApotcDataIds(seurat_obj)

getApotcRunIds(seurat_obj)
```

Arguments

`seurat_obj`	a seurat object that has had RunAPOTC ran on it before in order of the functions being called.

Value

a character vector of all run ids of previous RunAPOTC runs, in the order they were ran in. If there are no runs on the object, it returns `NULL`.

Examples

```
pbmc <- RunAPOTC(
    seurat_obj = get(data("combined_pbmc")),
    reduction_base = "umap",
    clonecall = "strict",
    verbose = FALSE
)

getApotcDataIds(pbmc)
#> [1] "umap;CTstrict;_;_"

pbmc <- RunAPOTC(
    seurat_obj = pbmc,
    reduction_base = "umap",
    clonecall = "gene",
    verbose = FALSE
)

getApotcDataIds(pbmc)
#> [1] "umap;CTstrict;_;_" "umap;CTgene;_;_"
```

`getLastApotcDataId` *Get the object id of the most recent RunAPOTC run on a seurat object*

Description

[Stable]

A convenience function to get the object id of the most recent valid `RunAPOTC()` run, to be used by `APOTCPlot()` and AdjustAPOTC

Usage

```
getLastApotcDataId(seurat_obj)

getLastApotcRunId(seurat_obj)
```

Arguments

`seurat_obj`	a seurat object that has had RunAPOTC ran on it before in order of the functions being called.

Value

a character of the object id of the last `RunAPOTC()` call

Examples

```
# first run
pbmc <- RunAPOTC(
    seurat_obj = get(data("combined_pbmc")),
    reduction_base = "umap",
    clonecall = "strict",
    verbose = FALSE
)

getLastApotcDataId(pbmc)
#> [1] "umap;CTstrict;_;_"

# second run with a different clonecall
pbmc <- RunAPOTC(
    seurat_obj = pbmc,
    reduction_base = "umap",
    clonecall = "gene",
    verbose = FALSE
)

getLastApotcDataId(pbmc)
#> [1] "umap;CTgene;_;_"
```

---

getReductionCentroids *Calculate seurat cluster centroids based on a Dimensional reduction*

Description

[Stable]

Utility function to calculate the physical xy coordinates of each seurat cluster based on a dimensional reduction already present in the object. The results are returned in a list with the length of the number of distinct seurat clusters based on the seurat_obj `meta.data`.

Usage

```
getReductionCentroids(seurat_obj, reduction)
```

Arguments

`seurat_obj`	input seurat object with the dimensional reduction of choice already present, and seurat clusters computed.
`reduction`	character. The reduction that the centroid calculation should be based on.

Value

A list of the length of the number of distinct clusters in the seurat object metadata, where each element of the list is a numeric vector of length 2, with the numbers corresponding to the x and y coordinate respectively of the seurat cluster with the corresponding index.

Examples

```
data("combined_pbmc")
getReductionCentroids(combined_pbmc, reduction = "umap")
```

`getSharedClones` *Compute a list of clonotypes that are shared between seurat clusters*

Description

[Stable]

This function allows users to get a list of clonotypes that are shared between clusters based on the levels of the active cell identities / some custom identity based on the `alt_ident`. A list is returned with its **names** being the shared clonotypes, and the values are numeric vectors indicating the index of the clusters that clonotype is found in. The index corresponds to the index in the default levels of the factored identities.

If `run_id` is inputted, then the function will attempt to get the shared clonotypes from the corresponding APackOfTheClones run generated from `RunAPOTC()`. Otherwise, it will use the filtering / subsetting parameters to generate the shared clones.

Usage

```
getSharedClones(
  seurat_obj,
  reduction_base = "umap",
  clonecall = "strict",
  ...,
  extra_filter = NULL,
  alt_ident = NULL,
  run_id = NULL,
  top = NULL,
  top_per_cl = NULL,
  intop = NULL,
  intop_per_cl = NULL,
  publicity = c(2L, Inf)
)
```

Arguments

`seurat_obj`	Seurat object with one or more dimension reductions and already have been integrated with a TCR/BCR library with `scRepertoire::combineExpression`.
`reduction_base`	character. The seurat reduction to base the clonal expansion plotting on. Defaults to `'umap'` but can be any reduction present within the reductions slot of the input seurat object, including custom ones. If ''pca'', the cluster coordinates will be based on PC1 and PC2. However, generally APackOfTheClones is used for displaying UMAP and occasionally t-SNE versions to intuitively highlight clonal expansion.

clonecall — character. The column name in the seurat object metadata to use. See scRepertoire documentation for more information about this parameter that is central to both packages.

... — additional "subsetting" keyword arguments indicating the rows corresponding to elements in the seurat object metadata that should be filtered by. E.g., seurat_clusters = c(1, 9, 10) will filter the cells to those in the seurat_clusters column with any of the values 1, 9, and 10. Unfortunately, column names in the seurat object metadata cannot conflict with the keyword arguments. **MAJOR NOTE** if any subsetting keyword arguments are a *prefix* of any preceding argument names (e.g. a column named reduction is a prefix of the reduction_base argument) R will interpret it as the same argument unless *both* arguments are named. Additionally, this means any subsequent arguments *must* be named.

extra_filter — character. An additional string that should be formatted *exactly* like a statement one would pass into dplyr::filter that does *additional* filtering to cells in the seurat object - on top of the other keyword arguments - based on the metadata. This means that it will be logically AND'ed with any keyword argument filters. This is a more flexible alternative / addition to the filtering keyword arguments. For example, if one wanted to filter by the length of the amino acid sequence of TCRs, one could pass in something like extra_filter = "nchar(CTaa) - 1 > 10". When involving characters, ensure to enclose with single quotes.

alt_ident — character. By default, cluster identity is assumed to be whatever is in Idents(seurat_obj), and clones will be grouped by the active ident. However, alt_ident could be set as the name of some column in the meta data of the seurat object to be grouped by. This column is meant to have been a product of Seurat::StashIdent or manually added.

run_id — character. This will be the ID associated with the data of a run, and will be used by other important functions like APOTCPlot() and AdjustAPOTC. Defaults to NULL, in which case the ID will be generated in the following format:
reduction_base;clonecall;keyword_arguments;extra_filter
where if keyword arguments and extra_filter are underscore characters if there was no input for the ... and extra_filter parameters.

top — integer or numeric in (0, 1) - if not null, filters the output clones so that only the shared clonotypes with counts the top top count / proportion (for numeric in (0, 1) input) shared clones are kept. For cases where several clonotypes tie in size, the clonotype(s) added are not guaranteed but deterministic given the other arguments are identical.

top_per_cl — integer or numeric in (0, 1) - if not null, filters the output clones so that for each seurat cluster, only the clonotypes with the top_per_cl frequency/count is preserved when aggregating shared clones, in the same way as the above. Note that if inputted in conjunction with top, it will get the *intersection* of the clonotypes filtered each way. For cases where several clonotypes tie in size, the clonotype(s) added are not guaranteed but deterministic given the other arguments are identical.

intop — integer or numeric in (0, 1) - if not null, filters the raw clone sizes before computing the shared clonotypes so that only the clonotypes that have their overall size in the top intop largest sizes (if it is integer, else the intop proportion) are

kept. To emphasize, this argument **does not necessarily** return the `top` shared clones and likely a little less, because this filters the raw clone sizes, of which, its very likely that not all those clones end up being shared.

`intop_per_cl` integer or numeric in (0, 1) - if not null, filters the raw *clustered* clone sizes before computing shared clones, so that for every clone in a seurat cluster, the top `intop_per_cl` count / proportion (for numeric in (0, 1) input) clones are kept.

`publicity` numeric pair. A simple filter range of `c(lowerbound, upperbound)` to retain only shared clones with their "publicity" - number of clusters they are present in - within this range.

Value

a named list where each name is a clonotype, each element is a numeric indicating which seurat cluster(s) its in, in no particular order. If no shared clones are present, the output is an empty list.

Examples

```
data("combined_pbmc")

getSharedClones(combined_pbmc)

getSharedClones(
    combined_pbmc,
    orig.ident = c("P17B", "P18B"), # a named subsetting parameter
    clonecall = "aa"
)

# extract shared clones from a past RunAPOTC run
combined_pbmc <- RunAPOTC(
    combined_pbmc, run_id = "foo", verbose = FALSE
)

getSharedClones(
    combined_pbmc, run_id = "foo", top = 5
)

# doing a run and then getting the clones works too
combined_pbmc <- RunAPOTC(combined_pbmc, run_id = "run1", verbose = FALSE)
getSharedClones(combined_pbmc, run_id = "run1")
```

`mergeCloneSizes` *Merge a list of Clustered Clonotype Frequency Tables*

Description

[Experimental]

The list of clustered clonotype frequencies from countCloneSizes can be merged by this function to a frequency table of all clonotypes similar to the data that can be seen in the seurat object metadata. By default, this function sorts the table with largest clonotypes first, and this may be useful for quickly gauging which clonotypes are the most expanded overall.

Usage

```
mergeCloneSizes(clustered_clone_sizes, sort_decreasing = TRUE)
```

Arguments

`clustered_clone_sizes`	the output of countCloneSizes.
`sort_decreasing`	a logical or NULL. If `TRUE`/`FALSE`, sorts the table by clonotype frequency with largest/smallest clones first, and if NULL, no order is guaranteed but the output is deterministic.

Value

a table object

See Also

countCloneSizes

Examples

```
clustered_clone_sizes <- countCloneSizes(get(data("combined_pbmc")))
mergeCloneSizes(clustered_clone_sizes)
```

`overlayLegend` *overlay a clone size legend on an APackOfTheClones plot*

Description

[Stable]

This function has most of the parameters related to legend in `APOTCPlot()`, and can plot a new / override the current legend. However, it is very important that the input plot to the function is a plot generated solely by `APOTCPlot()` or `vizAPOTC()` due to it being a custom ggplot object. It will not override or erase any additional layers that the user/other functions have added. To just remove the legend, see removeLegend.

Usage

```
overlayLegend(
  apotc_ggplot,
  legend_sizes = "auto",
  legend_position = "auto",
  legend_buffer = 0.2,
  legend_color = "#808080",
  legend_spacing = "auto",
  legend_label = "Clone sizes",
  legend_text_size = 5,
  add_legend_background = TRUE,
  add_legend_centerspace = 0,
  linetype = "blank",
  res = 360L
)
```

Arguments

`apotc_ggplot`	a ggplot object that is the output of `APOTCPlot()` or `vizAPOTC()`
`legend_sizes`	numeric vector. Indicates the circle sizes to be displayed on the legend, and will always be sorted from smallest to greatest. Defaults to `"auto"` which estimate a reasonable range of sizes to display.
`legend_position`	character or numeric. Can be set to either `"top_left"`, `"top_right"`, `"bottom_left"`, `"bottom_right"` and places the legend roughly in the corresponding position. Otherwise, can be a numeric vector of length 2 indicating the x and y position of the *topmost (smallest) circle* of the legend.
`legend_buffer`	numeric. Indicates how much to "push" the legend towards the center of the plot from the selected corner. If negative, will push away
`legend_color`	character. Indicates the hex color of the circles displayed on the legend. Defaults to the hex code for a gray tone
`legend_spacing`	numeric. Indicates the horizontal distance between each stacked circle on the size legend. Defaults to `"auto"` which will use an estimated value depending on plot size
`legend_label`	character. The title of the legend, which defaults to `"clone sizes`.
`legend_text_size`	numeric. The text size of the letters and numbers on the legend
`add_legend_background`	logical. If `TRUE`, will add a border around the legend and fill the background to be white, overlaying anything else.
`add_legend_centerspace`	numeric. An additional amount of distance changed between the circle sizes on the left side of the legend and the numbers on the right. Useful to set to around 0.5 (or more / less) when there are particularly large clone sizes that may cover the numbers.

`linetype`	The type of outline each circle should have. defaults to `"blank` meaning no outline. More information is in the function documentation of `ggforce::geom_circle`.
`res`	The number of points on the generated path per full circle. From plot viewers, if circles seem slightly too pixelated, it is recommended to first try to export the plot as an `.svg` before increasing `res` due to increased plotting times from ggforce::geom_circle.

Details

The size legend on APackOfTheClones plots are simply a collection of annotation layers of the `rect`, `text`, `circle` geoms. Therefore it isn't quite a ggplot legend. In the actual ggplot object, the `$layers` element should all be named with an empty character `""`, and those that comprise the annotation layers of the legend should be named whatever `APackOfTheClones:::.ApotcLegendLayerName` is. Note that this is simply an implementation detail that the user should not interfere with.

Value

A ggplot object of the APackOfTheClones clonal expansion plot of the seurat object. There is an additional 10th element in the object named `"APackOfTheClones"` used by other functions in this package and shouldn't interfere with any other ggplot functionality. (As far as currently known)

See Also

removeLegend

Examples

```
library(dplyr)

# create a plot with a legend
apotc_plot <- vizAPOTC(get(data("combined_pbmc")), verbose = FALSE)

# reposition the legend to top right
overlayLegend(apotc_plot, legend_position = "top right")

# use different sizes and label - may be nice to use the pipe
apotc_plot %>% overlayLegend(
    legend_sizes = c(1, 3, 7, 9),
    legend_label = "odd sizes"
)
```

`removeLegend` *Remove current APackOfTheClones legend*

Description

[Stable]

Removes the clone size legend on an APackOfTheClones plot, if one is present. Will preserve any additional ggplot layers.

Usage

```
removeLegend(apotc_ggplot)
```

Arguments

apotc_ggplot	a ggplot object that is the output of APOTCPlot() or vizAPOTC()

Value

A ggplot object of the APackOfTheClones clonal expansion plot of the seurat object. There is an additional 10th element in the object named "APackOfTheClones" used by other functions in this package and shouldn't interfere with any other ggplot functionality. (As far as currently known)

See Also

overlayLegend

Examples

```
# create an APackOfTheClones plot with a legend
apotc_plot <- vizAPOTC(
    get(data("combined_pbmc")),
    add_size_legend = TRUE,
    verbose = FALSE
)

# remove the legend
apotc_plot <- removeLegend(apotc_plot)
apotc_plot
```

---

renameApotcRun *Rename an APackOfTheClones run*

Description

[Stable]

A function to rename an APackOfTheClones run identified by its run id in a Seurat object.

Usage

```
renameApotcRun(seurat_obj, old_run_id, new_run_id)
```

Arguments

seurat_obj	A Seurat object containing APackOfTheClones data - the output of RunAPOTC()
old_run_id	Character. The current id of the APackOfTheClones run to be renamed.
new_run_id	Character. The new id to assign to the APackOfTheClones run.

Value

A Seurat object with the APackOfTheClones run renamed.

Examples

```
pbmc <- RunAPOTC(
    seurat_obj = get(data("combined_pbmc")),
    reduction_base = "umap",
    clonecall = "strict",
    run_id = "run1",
    verbose = FALSE
)

pbmc <- renameApotcRun(pbmc, "run1", "new_run")
# Now "run1" has been renamed to "new_run"
```

RunAPOTC *Run the APackOfTheClones method on a combined Seurat object for downstream visualization of clonal expansion*

Description

[Stable]

Computes necessary information for an APackOfTheClones clonal expansion plot (APOTCPlot()) and stores it in the seurat object. Gets sizes of unique clones and utilizes a circle-packing algorithm to pack circles representing individual clones in approximately the same dimensional reduction (reduction_base) coordinates based on some cell ident (defaults to the active ident).

The parameter extra_filter along with an unlimited number of additional keyword arguments can be used to filter the cells by certain conditions in the metadata, and new results will be stored in addition to other runs the users may have done.

Each APackOfTheClones run is uniquely identified by the parameters reduction_base, clonecall, extra_filter, and any additional keywords passed to filter the metadata. Each distinct run result is stored in the seurat object and has an associated Id generated from the aforementioned parameters. To view the id of the latest run, call getLastApotcDataId. To view all the ids of previous runs, call getApotcDataIds. To work further with a specific run (most importantly, plotting), the user can use this id in the arguments with is slightly more convenient than passing in the original RunAPOTC parameters again but both ways work.

If the user wishes to manually customize/fix the expansion plot generated, the circular packing information can be modified with the AdjustAPOTC function.

Usage

```
RunAPOTC(
  seurat_obj,
  reduction_base = "umap",
```

```
  clonecall = "strict",
  ...,
  extra_filter = NULL,
  alt_ident = NULL,
  run_id = NULL,
  clone_scale_factor = "auto",
  rad_scale_factor = 0.95,
  order_clones = TRUE,
  try_place = FALSE,
  repulse = TRUE,
  repulsion_threshold = 1,
  repulsion_strength = 1,
  max_repulsion_iter = 20L,
  override = FALSE,
  verbose = TRUE
)
```

Arguments

`seurat_obj`	Seurat object with one or more dimension reductions and already have been integrated with a TCR/BCR library with `scRepertoire::combineExpression`.
`reduction_base`	character. The seurat reduction to base the clonal expansion plotting on. Defaults to `'umap'` but can be any reduction present within the reductions slot of the input seurat object, including custom ones. If ''pca'', the cluster coordinates will be based on PC1 and PC2. However, generally APackOfTheClones is used for displaying UMAP and occasionally t-SNE versions to intuitively highlight clonal expansion.
`clonecall`	character. The column name in the seurat object metadata to use. See `scRepertoire` documentation for more information about this parameter that is central to both packages.
`...`	additional "subsetting" keyword arguments indicating the rows corresponding to elements in the seurat object metadata that should be filtered by. E.g., `seurat_clusters = c(1, 9, 10)` will filter the cells to those in the `seurat_clusters` column with any of the values 1, 9, and 10. Unfortunately, column names in the seurat object metadata cannot conflict with the keyword arguments. **MAJOR NOTE** if any subsetting keyword arguments are a *prefix* of any preceding argument names (e.g. a column named `reduction` is a prefix of the `reduction_base` argument) R will interpret it as the same argument unless *both* arguments are named. Additionally, this means any subsequent arguments *must* be named.
`extra_filter`	character. An additional string that should be formatted *exactly* like a statement one would pass into dplyr::filter that does *additional* filtering to cells in the seurat object - on top of the other keyword arguments - based on the metadata. This means that it will be logically AND'ed with any keyword argument filters. This is a more flexible alternative / addition to the filtering keyword arguments. For example, if one wanted to filter by the length of the amino acid sequence of TCRs, one could pass in something like `extra_filter = "nchar(CTaa) - 1 > 10"`. When involving characters, ensure to enclose with single quotes.

`alt_ident` character. By default, cluster identity is assumed to be whatever is in `Idents(seurat_obj)`, and clones will be grouped by the active ident. However, `alt_ident` could be set as the name of some column in the meta data of the seurat object to be grouped by. This column is meant to have been a product of `Seurat::StashIdent` or manually added.

`run_id` character. This will be the ID associated with the data of a run, and will be used by other important functions like `APOTCPlot()` and AdjustAPOTC. Defaults to `NULL`, in which case the ID will be generated in the following format: `reduction_base;clonecall;keyword_arguments;extra_filter`

where if keyword arguments and extra_filter are underscore characters if there was no input for the `...` and `extra_filter` parameters.

`clone_scale_factor`
Dictates how much to scale each circle(between 0,1) radius when converting from clonotype counts into circles that represent individual clonotypes. The argument defaults to the character `"auto"`, and if so, the most visually pleasing factor will be estimated.

`rad_scale_factor`
numeric between 0 and 1. This value decreases the radius of the smallest clones by this scale factor. And the absolute value of this decrease will be applied to all packed circles, effectively shrinking all circles on the spot, and introduce more constant spacing in between.

`order_clones` logical. Decides if the largest clone circles should be near cluster centroids. This is highly recommended to be set to TRUE for increased intuitiveness of the visualization, as resulting plots tend to give an improved impression of the proportion of expanded clones. If `FALSE`, will randomly scramble the positions of each circle. For the sake of being replicable, a random seed is recommended to be set with set.seed.

`try_place` If `TRUE`, always minimizes distance from a newly placed circle to the origin in the circle packing algorithm.

`repulse` If `TRUE`, will attempt to push overlapping clusters away from each other.

`repulsion_threshold`
numeric. The radius that clonal circle clusters overlap is acceptable when repulsing.

`repulsion_strength`
numeric. The smaller the value the less the clusters repulse each other per iteration, and vice versa.

`max_repulsion_iter`
integer. The number of repulsion iterations.

`override` logical. If `TRUE`, will override any existing APackOfTheClones run data with the same `run_id`.

`verbose` logical. Decides if visual cues are displayed to the R console of the progress.

Details

Note that the subsetting arguments `...` and `extra_filter` are only a quick convenience to subset based on metadata, and the `subset` S3 method defined in `Seurat` is much more mature are has more

features. Additionally, users need to work with data subsets are recommended to and likely already are working with seurat objects subsetted/split with `Seurat::SplitObject`.

All APackOfTheClones run data is stored in the Seurat object under `seurat_object@misc$APackOfTheClones`, which is a list of S4 objects of the type "ApotcData", with each element corresponding to a unique run. The id of each run is the name of each element in the list. The user **really shouldn't** manually modify anything in the list as it may cause unexpected behavior with many other functions.

Additionally, it logs a seurat command associated with the run in the `seurat_object@commands` slot as a "SeuratCommand" object (from Seurat), where the name of the object in the list is formatted as `RunAPOTC.run_id`.

Value

A modified version of the input seurat object, which harbors data necessary for visualizing the clonal expansion of the cells with `APOTCPlot()` and has a friendly user interface to modify certain attributes with AdjustAPOTC.

Cluster labelling

For the ident that was used to cluster the clones, labels for each cluster are inferred and stored in the run so that they can be used by other functions and optionally overlaid on the plot over clusters. If the levels of the ident used is a naturally ordered integer sequence, then the labels generated would be `"C1", "C2", "C3" ...` , else they would be the actual ident levels themselves.

See Also

`APOTCPlot()`, AdjustAPOTC, getApotcDataIds

Examples

```
data("combined_pbmc")

# this is the recommended approach to use a custom run_id with default params
combined_pbmc <- RunAPOTC(combined_pbmc, run_id = "default", verbose = FALSE)

# here's a seperate run with some filters to the meta data, where
# `orig.ident` is a custom column in the example data. Notice that it is not
# a `RunAPOTC()` parameter but a user keyword argument
combined_pbmc <- RunAPOTC(
    combined_pbmc, run_id = "sample17", orig.ident = c("P17B", "P17L"),
    verbose = FALSE
)

# the exact same thing can be achieved with the `extra_filter` parameter
combined_pbmc <- RunAPOTC(
    combined_pbmc,
    run_id = "sample17",
    extra_filter = "substr(orig.ident, 2, 3) == '17'",
    override = TRUE,
    verbose = FALSE
)
```

showCloneHighlight *Highlight specific clones on an APackOfTheClones ggplot*

Description

[Experimental]

This is an analogue for `scRepertoire::highlightClones` that can highlight certain clonotypes on an APackOfTheClones clonal expansion plot. For most combinations of the arguments, there will be a ggplot fill legend on the right side that correspond to each (existing) clonotype.

Usage

```
showCloneHighlight(
  apotc_ggplot,
  clonotype,
  color_each = TRUE,
  default_color = "#808080",
  scale_bg = 1,
  fill_legend = TRUE
)
```

Arguments

`apotc_ggplot`	A ggplot object that is the output of `APOTCPlot()` or `vizAPOTC()` of an APackOfTheClones plot to be highlighted on.
`clonotype`	character vector of the sequence(s) to highlight. Note that it must be of the clonecall of the code that created the plot. A warning will be shown if any of the sequences are not present.
`color_each`	Either a logical of length 1, or a character(s). It is `TRUE` by default, which assigns a unique default ggplot color to each highlighted clone. If `FALSE`, each highlighted clone will retain its current color and no legend based on color is shown. A possible application here is to simply gauge the distribution of any shared clone. It can also indicate the uniform color of each highlighted clone: if it is a character of length 1, all highlighted clones will be of that color. Else it must be a character vector of the same length as `clonotype`, with each color corresponding to the clone. Here is a suitable place to use any palette function from the many other CRAN palette packages such as `"viridis"` or `"RColorBrewer"`. Note that currently, the user must ensure `clonotype` contains only unique characters.
`default_color`	A character of length 1 or `NULL` indicating the color of non-highlighted clones. If `NULL`, all un-highlighted sequences will retain their original color in `sc.data`. Else, if it is a character, it should be a valid color that all un-highlighted clones are. Defaults to the hex code for gray.
`scale_bg`	A positive numeric. Scales the brightness value of each color of the non-highlighted clones by itself as a scaling factor. Defaults to 1 which will not alter the current

	brightness. Note that if `color_each = FALSE` and `default_color = NULL`, this is equivalent to not highlighting any clones - in this case, it may be useful to alter `scale_bg` slightly so that the non-highlighted clones are darkened/brightened.
`fill_legend`	logical indicating whether a ggplot legend of the "fill" of each clonotype should be displayed.

Details

Under the hood, this function simply mutates the plotting data.frame under `$data` in the ggplot object, and operates on a column named `color`.

Note that if `color_each = FALSE` and `default_color = NULL`, this is equivalent to simply not highlighting anything and a warning will be shown.

Value

A ggplot object with the data modified to the highlighted colors

Examples

```
library(dplyr)
data("combined_pbmc")

# piping the plot can be nice to read syntactically -
# By default, assigns unique colors to highlights and everything else is gray
vizAPOTC(combined_pbmc, clonecall = "aa", verbose = FALSE) %>%
    showCloneHighlight("CASLSGSARQLTF_CASSSTVAGEQYF")

# one useful application is to highlight shared clones - beware that the
# clonotype sequences may get extremely long in the legend
shared_aa_clones <- names(getSharedClones(combined_pbmc, clonecall = "aa"))
vizAPOTC(combined_pbmc, clonecall = "aa", verbose = FALSE) %>%
    showCloneHighlight(shared_aa_clones)
```

---

updateApotc *Update APackOfTheClones objects to the latest format*

Description

This generic function updates objects created by APackOfTheClones to the latest format or structure required by the package. It supports Seurat objects and ggplot objects generated by APackOfTheClones functions.

The current possible changes made by the function are:

- when called on ggplots generated by `vizAPOTC()` / `APOTCPlot()`, layer names will be made unique.

Usage

```
updateApotc(x, verbose = TRUE, ...)
```

Arguments

x	An object to update. Supported types are Seurat objects and ggplot objects generated by APackOfTheClones.
verbose	Logical. Whether to print messages about the update process. Default is TRUE.
...	Additional arguments passed to methods. Currently unused.

Value

The updated object, or the original object if no update is required.

Examples

```
data("combined_pbmc")
apotc_plot <- vizAPOTC(combined_pbmc)
apotc_plot <- updateApotc(apotc_plot, verbose = TRUE)
apotc_plot
```

---

vizAPOTC *Directly visualize clonal expansion of a combined seurat object*

Description

[Stable]

This function combines the functionality of both `RunAPOTC()` and `APOTCPlot()`. Given a Seurat object, it first runs the APackOfTheClones method (`RunAPOTC()`) to compute clonal expansion information, and then generates a ggplot2 object of the clonal expansion plot with a circle size legend. (`APOTCPlot()`)

Usage

```
vizAPOTC(
  seurat_obj,
  reduction_base = "umap",
  clonecall = "strict",
  ...,
  extra_filter = NULL,
  alt_ident = NULL,
  clone_scale_factor = "auto",
  rad_scale_factor = 0.95,
  order_clones = TRUE,
  try_place = FALSE,
```

```
  repulse = TRUE,
  repulsion_threshold = 1,
  repulsion_strength = 1,
  max_repulsion_iter = 20L,
  show_shared = NULL,
  only_link = NULL,
  show_all_links = FALSE,
  clone_link_width = "auto",
  clone_link_color = "black",
  clone_link_alpha = 0.5,
  res = 360L,
  linetype = "blank",
  use_default_theme = TRUE,
  retain_axis_scales = FALSE,
  alpha = 1,
  show_labels = FALSE,
  label_size = 5,
  add_size_legend = TRUE,
  legend_sizes = "auto",
  legend_position = "auto",
  legend_buffer = 0.2,
  legend_color = "#808080",
  legend_spacing = "auto",
  legend_label = "Clone sizes",
  legend_text_size = 5,
  add_legend_background = TRUE,
  add_legend_centerspace = 0,
  detail = TRUE,
  verbose = TRUE
)
```

Arguments

seurat_obj	A seurat object that has been integrated with clonotype data with scRepertoire::combineExpression.
reduction_base	character. The seurat reduction to base the clonal expansion plotting on. Defaults to 'umap' but can be any reduction present within the reductions slot of the input seurat object, including custom ones. If ''pca'', the cluster coordinates will be based on PC1 and PC2. However, generally APackOfTheClones is used for displaying UMAP and occasionally t-SNE versions to intuitively highlight clonal expansion.
clonecall	character. The column name in the seurat object metadata to use. See scRepertoire documentation for more information about this parameter that is central to both packages.
...	additional "subsetting" keyword arguments indicating the rows corresponding to elements in the seurat object metadata that should be filtered by. E.g., seurat_clusters = c(1, 9, 10) will filter the cells to those in the seurat_clusters column with any of the values 1, 9, and 10. Unfortunately, column names in the seurat object metadata cannot conflict with the keyword arguments. **MAJOR NOTE** if any

subsetting keyword arguments are a *prefix* of any preceding argument names (e.g. a column named `reduction` is a prefix of the `reduction_base` argument) R will interpret it as the same argument unless *both* arguments are named. Additionally, this means any subsequent arguments *must* be named.

`extra_filter` character. An additional string that should be formatted *exactly* like a statement one would pass into dplyr::filter that does *additional* filtering to cells in the seurat object - on top of the other keyword arguments - based on the metadata. This means that it will be logically AND'ed with any keyword argument filters. This is a more flexible alternative / addition to the filtering keyword arguments. For example, if one wanted to filter by the length of the amino acid sequence of TCRs, one could pass in something like `extra_filter = "nchar(CTaa) - 1 > 10"`. When involving characters, ensure to enclose with single quotes.

`alt_ident` character. By default, cluster identity is assumed to be whatever is in `Idents(seurat_obj)`, and clones will be grouped by the active ident. However, `alt_ident` could be set as the name of some column in the meta data of the seurat object to be grouped by. This column is meant to have been a product of `Seurat::StashIdent` or manually added.

`clone_scale_factor`
Dictates how much to scale each circle(between 0,1) radius when converting from clonotype counts into circles that represent individual clonotypes. The argument defaults to the character `"auto"`, and if so, the most visually pleasing factor will be estimated.

`rad_scale_factor`
numeric between 0 and 1. This value decreases the radius of the smallest clones by this scale factor. And the absolute value of this decrease will be applied to all packed circles, effectively shrinking all circles on the spot, and introduce more constant spacing in between.

`order_clones` logical. Decides if the largest clone circles should be near cluster centroids. This is highly recommended to be set to TRUE for increased intuitiveness of the visualization, as resulting plots tend to give an improved impression of the proportion of expanded clones. If `FALSE`, will randomly scramble the positions of each circle. For the sake of being replicable, a random seed is recommended to be set with set.seed.

`try_place` If `TRUE`, always minimizes distance from a newly placed circle to the origin in the circle packing algorithm.

`repulse` If `TRUE`, will attempt to push overlapping clusters away from each other.

`repulsion_threshold`
numeric. The radius that clonal circle clusters overlap is acceptable when repulsing.

`repulsion_strength`
numeric. The smaller the value the less the clusters repulse each other per iteration, and vice versa.

`max_repulsion_iter`
integer. The number of repulsion iterations.

`show_shared` The output of getSharedClones can be inputted here, and the resulting plot will overlay lines between clone circles if that clonotype is common between clusters. Note that the input **must** be generated from data in the correct `APackOfTheClones`

run, and the behavior is undefined otherwise and will likely error. The next 4 arguments allow for aesthetic customization of these line links.

`only_link` Optional integer indicating to only display clone links originating from this cluster if showing shared clones.

`show_all_links` **[Experimental]** logical. If `TRUE`, will show all links between clusters. Defaults to `FALSE` which only shows links between some clusters without overlapping, although there is currently no way to control which.

`clone_link_width`
numeric. The width of the lines that connect shared clones. Defaults to `"auto"` which will estimate a reasonable value depending on circle sizes.

`clone_link_color`
character. The color of the lines that connect shared clones. Defaults to `"blend"` which will use the average colors of the two connected clones. Else, any hex color or valid color string input will work, and the corresponding color will be applied on all links.

`clone_link_alpha`
numeric. The alpha of the lines that connect shared clones.

`res` The number of points on the generated path per full circle. From plot viewers, if circles seem slightly too pixelated, it is recommended to first try to export the plot as an `.svg` before increasing `res` due to increased plotting times from ggforce::geom_circle.

`linetype` The type of outline each circle should have. defaults to `"blank` meaning no outline. More information is in the function documentation of `ggforce::geom_circle`.

`use_default_theme`
logical that defaults to `TRUE`. If `TRUE`, the resulting plot will have the same theme as the seurat reference reduction plot. Else, the plot will simply have a blank background.

`retain_axis_scales`
If `TRUE`, approximately maintains the axis scales of the original reduction plot. However, it will only attempt to extend the axes and never shorten. Users are recommended to set this to `TRUE` especially if working with subsetted versions of the clonal data to better preserve the geometric relation to the original dimensional reduction.

`alpha` numeric. The alpha of the circles in (0, 1]. Defaults to 1.

`show_labels` If `TRUE`, will label each circle cluster at the centroid, defaulting to "C0, C1, ...".

`label_size` The text size of labels if shown. Defaults to 5.

`add_size_legend`
If `TRUE`, adds a legend to the plot visualizing the relative sizes of clones. Note that it is simply an overlay and not a real ggplot2 legend.

`legend_sizes` numeric vector. Indicates the circle sizes to be displayed on the legend, and will always be sorted from smallest to greatest. Defaults to `"auto"` which estimate a reasonable range of sizes to display.

`legend_position`
character or numeric. Can be set to either `"top_left"`, `"top_right"`, `"bottom_left"`, `"bottom_right"` and places the legend roughly in the corresponding position.

Otherwise, can be a numeric vector of length 2 indicating the x and y position of the *topmost (smallest) circle* of the legend.

`legend_buffer` numeric. Indicates how much to "push" the legend towards the center of the plot from the selected corner. If negative, will push away

`legend_color` character. Indicates the hex color of the circles displayed on the legend. Defaults to the hex code for a gray tone

`legend_spacing` numeric. Indicates the horizontal distance between each stacked circle on the size legend. Defaults to `"auto"` which will use an estimated value depending on plot size

`legend_label` character. The title of the legend, which defaults to `"clone sizes`.

`legend_text_size`
numeric. The text size of the letters and numbers on the legend

`add_legend_background`
logical. If `TRUE`, will add a border around the legend and fill the background to be white, overlaying anything else.

`add_legend_centerspace`
numeric. An additional amount of distance changed between the circle sizes on the left side of the legend and the numbers on the right. Useful to set to around 0.5 (or more / less) when there are particularly large clone sizes that may cover the numbers.

`detail` logical. If `FALSE`, will only plot entire clusters as one large circle, which may be useful in cases where there are a high number of clones resulting in a large number of circles on the resulting ggplot, which has increased plotting times, and certain aspects of the plot needs to be finely adjusted with AdjustAPOTC or simply inspected. This should not be set to `FALSE` for the actual clonal expansion plot.

`verbose` logical. Decides if visual cues are displayed to the R console of the progress.

Details

Note that the subsetting arguments `...` and `extra_filter` are only a quick convenience to subset based on metadata, and the `subset` S3 method defined in `Seurat` is much more mature are has more features. Additionally, users need to work with data subsets are recommended to and likely already are working with seurat objects subsetted/split with `Seurat::SplitObject`.

Value

A ggplot object of the APackOfTheClones clonal expansion plot of the seurat object. There is an additional 10th element in the object named `"APackOfTheClones"` used by other functions in this package and shouldn't interfere with any other ggplot functionality. (As far as currently known)

Cluster labelling

For the ident that was used to cluster the clones, labels for each cluster are inferred and stored in the run so that they can be used by other functions and optionally overlaid on the plot over clusters. If the levels of the ident used is a naturally ordered integer sequence, then the labels generated would be `"C1", "C2", "C3" ...` , else they would be the actual ident levels themselves.

See Also

AdjustAPOTC()

Examples

```
data("combined_pbmc")

# plot with default parameters
vizAPOTC(combined_pbmc, verbose = FALSE)

# use arguments from RunAPOTC and APOTCPlot
vizAPOTC(
    combined_pbmc, try_place = TRUE, show_labels = TRUE, verbose = FALSE
)
```

Index

∗ **datasets**
combined_pbmc, 10

AdjustAPOTC, 2, *4*, *6*, *8–10*, *15*, *18*, *24*, *26, 27*, *34*
AdjustAPOTC(), *35*
APOTCPlot, 6
APOTCPlot(), *2*, *4*, *8*, *15*, *18*, *20, 21*, *23, 24*, *26–30*

combined_pbmc, 10
containsApotcRun, 11
countCloneSizes, 12, *20*

deleteApotcData, 13
dplyr::filter, *4*, *7*, *12*, *18*, *25*, *32*

getApotcDataIds, 14, *24*, *27*
getApotcRunIds *(*getApotcDataIds*)*, 14
getLastApotcDataId, 15, *24*
getLastApotcRunId *(*getLastApotcDataId*)*, 15
getReductionCentroids, 16
getSharedClones, *8*, 17, *32*
ggforce::geom_circle, *8*, *22*, *33*

mergeCloneSizes, *13*, 19

overlayLegend, 20, *23*

removeLegend, *20*, 22, *22*
renameApotcRun, 23
RunAPOTC, 24
RunAPOTC(), *2*, *6, 7*, *15*, *17*, *23*, *30*

set.seed, *26*, *32*
showCloneHighlight, 28

updateApotc, 29

vizAPOTC, 30
vizAPOTC(), *20, 21*, *23*, *28, 29*